Tool Building as a Path to "Superintelligence"

David Koplow Tomer Galanti

February 11, 2026

Abstract

The *Diligent Learner* framework suggests LLMs can achieve superintelligence via test-time search, provided a sufficient step-success probability γ. In this work, we design a benchmark to measure γ on logical out-of-distribution inference. We construct a class of tasks involving GF(2) circuit reconstruction that grow more difficult with each reasoning step, and that are, from an information-theoretic standpoint, impossible to reliably solve unless the LLM carefully integrates all of the information provided. Our analysis demonstrates that while the γ value for small LLMs declines superlinearly as depth increases, frontier models exhibit partial robustness on this task. Furthermore, we find that successful reasoning at scale is contingent upon precise tool calls, identifying tool design as a critical capability for LLMs to achieve general superintelligence through the Diligent Learner framework.

1 Introduction

The recent emergence of large-scale LLMs has made multi-step reasoning increasingly practical Wei et al. [2022], Wang et al. [2022], Fu et al. [2023], Kojima et al. [2022], especially when combined with inference-time search, tool use, and verification. Much of this progress has been enabled by post-training, including reinforcement learning and preference-optimization methods Ouyang et al. [2022], Shao et al. [2024]. Recent work on the *Diligent Learner*, suggests search through reasoning steps on problems of bounded depth could lead to "superintelgent" agents with our existing architectures [Shalev-Shwartz and Shashua, 2025b].

The viability of this framework hinges on a critical quantity: the stepwise success probability, denoted by γ. The central premise of Diligent Learning is that test-time search can scale effectively only if the model's proposal distribution preserves a non-vanishing probability of generating the correct subsequent step. However, a central question remains unresolved; *as reasoning unfolds over longer horizons on tasks does the stepwise success probability γ always remain larger than a positive constant, or are there categories of problems for which it catastrophically degrades with depth?*

If a successful reasoning chain requires knowledge that the LLM has never been exposed to, then the answer is trivial. However, when it comes to the

problem of general superintelligence, the important question is whether or not the model can solve out of distribution problems while reasoning about prior learned relationships. In this work, we design a benchmark to measure exactly this quality.

While plenty of benchmarks exist for studying the reasoning capabilities of different models, existing evaluations are inadequate for this goal. Many benchmarks score only final answers, allow multiple valid intermediate paths, or permit shortcuts where performance comes from pattern-matching labeled data or memorizing prior examples. Thus, "stepwise reasoning" is confounded with exploiting benchmark-specific regularities and pre-trained knowledge.

To rigorously test the Diligent Learner hypothesis, we introduce a benchmark that is adversarial to such shortcuts. We design a form of Boolean circuit reconstruction from data over GF(2). The model must predict successive terms in an Algebraic Normal Form (ANF) of the circuit. Each step g in the reasoning chain has a unique correct continuation. To find it, the model must combine two distinct inputs: *(i) The Prefix: The history of the circuit revealed so far and (ii) The Evidence: A new batch of step-specific data.*

To ensure that the model cannot cheat, we employ an *adversarial sampling oracle*. This oracle generates evidence that appears statistically random unless the solver conditions it on the prefix. Consequently, strategies that rely solely on pattern-matching of the data or memorizing of the history will fail. Only a diligent solver that integrates both sources can recover the next term.

We provide the models with a perfect oracle that prevents them from going down incorrect reasoning paths. This structure allows us to find an empirical upper-bound for γ_g and measure the affect of problem complexity and reasoning depth without need to fine-tune the model learn how to handle back-tracking.

When we apply this metric to current systems, we find that in smaller models, γ_g collapses as the reasoning depth increases. Frontier models, however, sustain a high γ_g over long horizons when using tool calls.

2 Related Work

Reasoning as search with verification. A long line of work treats reasoning as a search problem where a model proposes candidate steps or solutions and an evaluation signal filters them; in LLMs, this appears in chain-of-thought prompting [Wei et al., 2022, Wang et al., 2022], Tree-of-Thought search over partial reasoning states [Yao et al., 2023], and iterative propose–critique–revise agentic loops such as Reflexion [Shinn et al., 2023]. These approaches are often augmented with tool use [Schick et al., 2023, Hao et al., 2023, Parisi et al., 2022, Shi et al., 2025] and explicit scratchpads [Nye et al., 2021] to maintain intermediate state and enable efficiently checkable substeps. On the theory side, chain-of-thought can be formalized as a task decomposition that makes otherwise hard concept classes learnable in a PAC-style setting [Yang et al., 2025a, Joshi et al., 2025], and next-token predictors can be viewed as general-purpose learners under suitable conditions [Malach, 2024]. Closely related, the

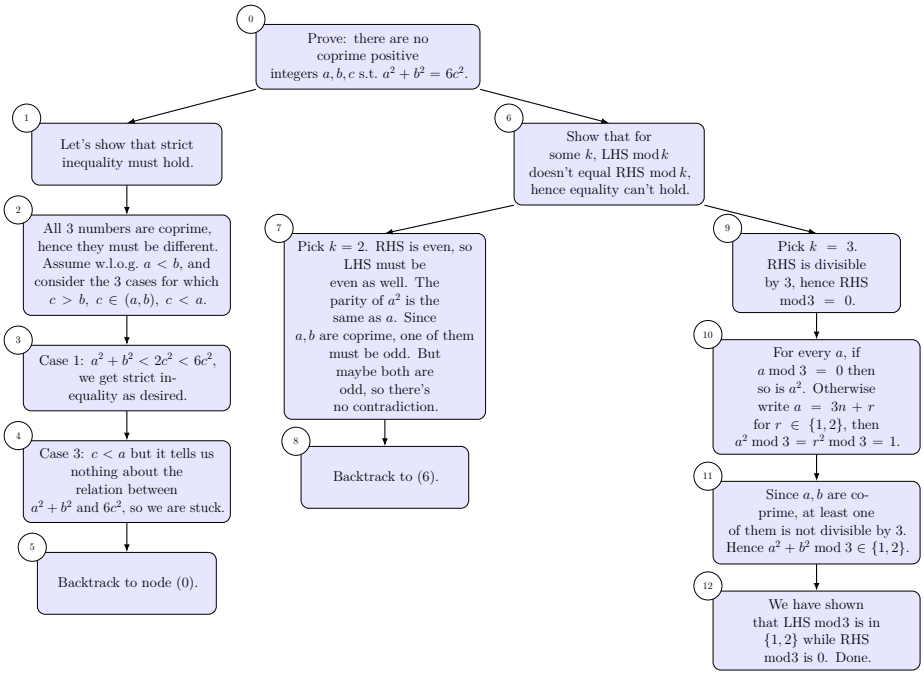

Figure 1: Diligent learner visualization from [Shalev-Shwartz and Shashua, 2025a].

LLM-ERM framework makes the propose-and-verify view explicit by treating the LLM as proposing hypotheses and a verifier as enforcing correctness [Singhal et al., 2025].

The Diligent Learner. Motivated by this broader view of reasoning as search guided by evaluation, a recent paper [Shalev-Shwartz and Shashua, 2025b] introduced the Diligent Learner framework in order to formalize the process as validator-guided depth-first search with an explicit BACKTRACK action. Its analysis isolates a single bottleneck, the per-step success probability γ, defined as the probability mass the policy assigns to "good" next steps that keep the current prefix completable. If γ does not collapse with depth and backtracking returns to the deepest correct prefix with high probability, search succeeds with controlled overhead [Shalev-Shwartz and Shashua, 2025b]. Our work targets the empirical gap left open by this theory: γ is defined abstractly, but there is no standard benchmark in which (i) the correct extension is unique at each step and (ii) shortcuts that ignore either the accumulated history or the fresh evidence are information-theoretically ineffective. We design such a benchmark so that γ_g is directly measurable as exact-next accuracy at depth g.

Benchmarks for testing reasoning in LLMs. Reasoning in LLMs is commonly evaluated via static, single-shot benchmarks spanning mathematical and logical problem solving, including grade-school word problems and multi-step

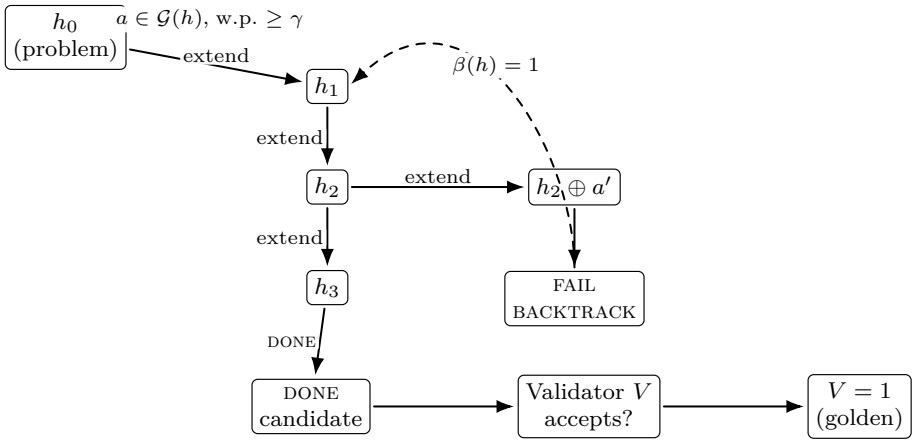

Figure 2: **Diligent Learner as validator-guided DFS.** Good extensions occur with probability at least γ. On failure, the policy backtracks to the deepest correct prefix $\beta(h)$ and continues search.

arithmetic [Cobbe et al., 2021], competition-style mathematics [Hendrycks et al., 2021], and curated hard subsets of broad capability suites [Suzgun et al., 2022]. A complementary line of work uses synthetic or controlled-distribution tasks to probe compositional generalization and algorithmic structure, such as formal mathematical expression generation [Frieder et al., 2025] and software engineering benchmarks [Jimenez et al., 2024, Yang et al., 2025b]. Finally, interactive evaluations target agentic reasoning with tools and environments, where success depends on multi-step planning and external actions (e.g., embodied or text-based worlds, or web interaction) [Shridhar et al., 2021, Zhou et al., 2024, Qin et al., 2024, Liu et al., 2025].

While these benchmarks are valuable for measuring end-to-end task performance, they typically (i) score only the final answer, (ii) admit many valid intermediate trajectories, and (iii) do not isolate whether success comes from the evolving history, the current evidence, or dataset-specific shortcuts. In contrast, our goal is to operationalize the Diligent Learner's per-step parameter γ in a setting where each depth g has a *unique* correct extension and where solvers that ignore either the revealed prefix or the fresh step-specific samples are information-theoretically ineffective. The stepwise GF(2) reconstruction benchmark we introduce is designed to meet these requirements, enabling direct measurement of γ_g as exact-next accuracy with a polynomial-time validator.

3 Background

3.1 The Diligent Learner

Reasoning as a search tree with validation. The Diligent Learner formalizes reasoning as building a rooted search tree whose root encodes the problem

instance, and whose nodes represent *semantic* reasoning states (partial chains). A root-to-leaf path corresponds to a proposed chain-of-thought (CoT). Leaves have two special types: DONE (a completed solution) and BACKTRACK (indicating a jump to an earlier node). A *validator* V checks a proposed step could logically follow from a prior one; a *golden path* is a root-to-DONE path accepted by V (see Figure 1).

Generator policy and step-success probability. Let V be a validator on the proposed next step. Let $\pi_\theta(\cdot \mid h, V)$ be a stochastic policy over valid actions given a partial reasoning state h (a prefix of a path). An *extension* action a proposes the next semantic step and S is a function that returns true if there exists a path to the conclusion of a reasoning problem given the prefix (h, a).

$$\Pr_{a \sim \pi_\theta(\cdot|h)} [a \in \{\forall a_i . S(h, a_i) = 1\}] \;\geq\; \gamma. \tag{1}$$

Intuitively, γ is the probability mass assigned to *useful* next moves that keep the trace completable.

Learned backtracking. The original diligent learner allows the model to make incorrect extensions so long as it is able to realize and revert via depth-first-search. However, in this paper we assume a very strong validator that prevents the model from going down incorrect paths so we never need to fine-tune the model to learn how to back-track.

The implications of γ. If the policy keeps a nontrivial chance γ of proposing a good next step and can backtrack to the last correct prefix, then depth-first search reaches a validator-accepted leaf with high probability without exponential blowup. Concretely, for a target failure probability δ and maximum depth $T_{\max}$, the analysis sets

$$O\left(T_{\max} \cdot \frac{\log(T_{\max}/\delta)}{\gamma}\right) \tag{2}$$

which is polynomial in $T_{\max}$ for constant γ [Shalev-Shwartz and Shashua, 2025b]. Thus, the central requirement is that γ not decay with depth; otherwise the search budget grows rapidly and the guarantee becomes vacuous.

4 Theory

The "Diligent Learner" hypothesis posits that a reasoning model can solve long multi-step problems by performing test-time search, as long as it maintains a non-vanishing probability γ of proposing a *good* next semantic step at every depth [Shalev-Shwartz and Shashua, 2025b]. We introduce a stepwise reconstruction benchmark to evaluate the limitations of this framework in which, at each step g, the model must extend a revealed partial solution (the current prefix) using a new batch of evidence specific to that step. The construction is designed to eliminate two shortcut strategies: (i) *data-only* prediction that ignores the prefix and tries to infer the next step from examples alone, and (ii) *history-only* prediction that ignores the new evidence and extrapolates from the prefix alone.

4.1 Problem formulation

We instantiate this in the reconstruction of Boolean functions over GF(2) under a fixed-prefix statistical obfuscation sampling oracle.

Targets in ANF over GF(2). Let $x = (a, v) \in \{0,1\}^{n+p}$ where $a = (a_1, \ldots, a_n) \in \{0,1\}^n$ are *address bits* and $v = (v_1, \ldots, v_p) \in \{0,1\}^p$ are *payload bits*. We consider Boolean targets represented in Algebraic Normal Form (ANF) as XORs of monomials $f(a, v) = \bigoplus_{j=1}^{n} t_j(a, v)$, where

$$t_j(a, v) := a_j \, M_j(v), \quad M_j(v) := \prod_{i \in S_j} v_i, \tag{3}$$

such that each support $S_j \subseteq [p]$ has fixed size $|S_j| = d - 1$. Thus each term has payload-degree $d-1$ and total degree d in (a, v). An *instance* is specified by the ordered sequence of supports $(S_1, \ldots, S_n)$ (equivalently, the ordered ANF terms $(t_1, \ldots, t_n)$), sampled once and then fixed.

Stepwise reconstruction game and step-success. At step $g \in \{0, \ldots, n-1\}$ the learner is given (i) the ordered prefix $P_g := (t_1, \ldots, t_g)$ and (ii) a fresh labeled sample set $\mathsf{S}_g := \{(x^{(k)}, y^{(k)})\}_{k=1}^{K}$ generated by the step-g oracle. The learner outputs a candidate monomial $\hat{t}$ and succeeds iff $\hat{t} = t_{g+1}$. We define the benchmark step-success probability:

$$\gamma_g \;:=\; \Pr_{\hat{t} \sim \pi_\theta(\cdot \mid P_g, \mathsf{S}_g)} \big[\hat{t} = t_{g+1}\big], \tag{4}$$

where the probability is over instance generation, oracle sampling, and model stochasticity. Because the instance commits to an ordered sequence, at depth g there is a unique correct extension t_{g+1}, so γ_g directly operationalizes step-success in our benchmark.

Estimator classes. We distinguish solvers by their information access: (i) **Diligent Estimator** ($\mathcal{A}_g$): access to (P_g, S_g); (ii) **Data-only Estimator** ($\mathcal{B}_g$): access to S_g but not P_g; (iii) **History-only Estimator** ($\mathcal{C}_g$): access to P_g but not S_g; (iv) **Partial Estimator** ($\mathcal{D}_g$): partial access to P_g and S_g. Let $\gamma_g^{\mathcal{X}}$ denote the exact-next success probability for class $\mathcal{X}$ at step g. Our benchmark is designed to enforce a separation of the form

$$\min_g \gamma_g^{\mathcal{A}} \;\geq\; Q \quad \text{while} \quad \gamma_g^{\mathcal{B}}, \gamma_g^{\mathcal{C}}, \gamma_g^{\mathcal{D}} \;\approx\; \tfrac{1}{\binom{p}{d-1}}. \tag{5}$$

for a nontrivial constant $1 \geq Q \gg 0$.

4.2 The statistical obfuscation construction

We now define the distribution over benchmark instances and the step-g sampling oracle. At step g, the label is the XOR of the unknown next payload monomial $M_{g+1}(v)$ and a randomized mask computable from the revealed prefix. Consequently, the labeled samples carry essentially no information about M_{g+1} unless the solver uses the prefix to cancel the mask.

Instance generation. Sample supports $S_1, \ldots, S_n \subseteq [p]$ with $|S_j| = d - 1$ once per instance (for example, i.i.d. uniform over $\{S \subseteq [p] : |S| = d - 1\}$,

or without replacement). The resulting instance fixes the ordered ANF terms $(t_1, \ldots, t_n)$ in (3).

Payload distribution (fixed weight). Fix a Hamming weight w and sample payloads uniformly from the sphere $\{v \in \{0,1\}^p : \|v\|_0 = w\}$. For any fixed S with $|S| = d-1$, define

$$\rho(w) \;:=\; \Pr_v\big[M_S(v) = 1\big] \;=\; \frac{\binom{w}{d-1}}{\binom{p}{d-1}} \quad (w \geq d-1), \tag{6}$$

and choose $w^\star$ to make $\rho(w)$ as close to $1/2$ as possible (so monomial evaluations are nearly balanced).

Step-g sampling oracle (fixed-prefix obfuscation). Given an instance and depth $g \in \{0, \ldots, n-1\}$, the oracle returns $\mathsf{S}_g = \{(a^{(k)}, v^{(k)}, y^{(k)})\}_{k=1}^K$ by sampling i.i.d. examples as follows: **(i)** Set $a_{g+1} = 1$ and set $a_j = 0$ for all $j > g+1$; **(ii)** Sample $a_1, \ldots, a_g \overset{i.i.d.}{\sim}$ Bernoulli(1/2); **(iii)** Sample v uniformly from $\{v : \|v\|_0 = w^\star\}$; **(iv)** Output $y := f(a, v)$. Under this oracle,

$$y \;=\; \underbrace{\Big(\bigoplus_{j=1}^{g} a_j\, M_j(v)\Big)}_{\text{prefix obfuscation}} \oplus \quad \underbrace{M_{g+1}(v)}_{\text{next-term signal}}\;, \tag{7}$$

since $a_{g+1} = 1$ and $a_j = 0$ for all $j > g+1$. The obfuscation prefix is computable from (P_g, x), but without P_g it behaves like a high-entropy statistical obfuscator that statistically obfuscates the signal.

4.3 Theoretical guarantees

We formalize two properties: (i) *per-sample obfuscation*, meaning that after marginalizing over the unknown prefix supports, each labeled example provides at most exponentially small Bayes advantage about the next-term signal to a solver that does not condition on the revealed prefix, and (ii) *recoverability*, meaning that a solver with access to the revealed prefix can cancel the mask and recover the next term in polynomial time from a fresh batch of samples.

Monomial firing at fixed weight. Because we draw payloads uniformly from a fixed Hamming sphere, the probability that a degree-$(d-1)$ payload monomial evaluates to 1 depends only on (p, w, d) and admits a simple closed form. This lets us choose $w^\star$ so that monomial evaluations are approximately balanced in expectation, which in turn makes the obfuscation effect high-entropy and prevents trivial leakage from biased labels.

Lemma 4.1 (Monomial firing probability at fixed Hamming weight)**.** *Fix integers $p \geq d-1 \geq 1$ and $w \in \{0, \ldots, p\}$. Let v be uniform over the Hamming sphere $\{v \in \{0,1\}^p : \|v\|_0 = w\}$, and fix any $S \subseteq [p]$ with $|S| = d-1$. Define*

$M_S(v) := \prod_{i \in S} v_i$. *Then*

$$\Pr\left[M_S(v) = 1\right] = \begin{cases} \frac{\binom{w}{d-1}}{\binom{p}{d-1}} = \frac{\binom{p-(d-1)}{w-(d-1)}}{\binom{p}{w}} & \text{if } w \geq d-1, \\ 0 & \text{if } w < d-1. \end{cases}$$

Per-sample obfuscation and shortcut resistance. A data-only estimator $\mathcal{B}_g$ observes sample triples $(a^{(k)}, v^{(k)}, y^{(k)})$ but not the revealed prefix P_g, and thus does not know the hidden prefix supports $(S_1, \ldots, S_g)$ that define the mask term. We do *not* claim a full information-theoretic impossibility for data-only solvers with K samples, since the same hidden supports are reused across all samples at a given step. Instead, we formalize a *per-sample* masking guarantee: after marginalizing over the unknown prefix supports, each labeled example has exponentially small Bayes advantage about the next-term signal unless one conditions on the prefix. Consequently, beating chance from data alone requires exploiting multi-sample structure to jointly infer hidden supports, which is empirically ineffective for our shortcut baselines at the benchmark sample sizes.

For a single example (a, v, y) at step g, define the signal bit $b := M_{g+1}(v)$ and the (unknown-to-$\mathcal{B}_g$) mask bit

$$B(a, v) \;:=\; \bigoplus_{j=1}^{g} a_j \, M_j(v) \;=\; \bigoplus_{j=1}^{g} a_j \, \mathbf{1}\{S_j \subseteq \mathrm{supp}(v)\}.$$

Let $m(a) := \sum_{j=1}^{g} a_j$ be the number of active prefix address bits in the example.

Lemma 4.2 (Bayes masking given observed (a, v))**.** *Assume the instance distribution samples $S_1, \ldots, S_g, S_{g+1}$ i.i.d. uniformly from $\{S \subseteq [p] : |S| = d-1\}$, independently of the oracle samples. Fix a step g and condition on a realized example (a, v) with $\|v\|_0 = w^\star$. Let*

$$\rho := \rho(w^\star) = \Pr_S\left[M_S(v) = 1\right] = \frac{\binom{w^\star}{d-1}}{\binom{p}{d-1}}, \quad m := m(a).$$

Then, marginalizing over the unknown prefix supports $(S_1, \ldots, S_g)$, for each $r \in \{0, 1\}$,

$$\Pr\left[B(a, v) = r \mid a, v\right] \;=\; \tfrac{1}{2}\left[1 + (-1)^r (1 - 2\rho)^m\right].$$

Moreover, $B(a, v)$ is independent of $b = M_{g+1}(v)$ given (a, v), and since $y = B(a, v) \oplus b$ we have

$$\left|\Pr[y = b \mid a, v] - \tfrac{1}{2}\right| \;=\; \tfrac{1}{2}|1 - 2\rho|^m.$$

Lem. 4.2 quantifies *single-sample* leakage. After averaging over the unknown prefix supports $(S_1, \ldots, S_g)$, each label can be written as $y = b \oplus B(a, v)$, where $b := M_{g+1}(v)$, such that the mask $B(a, v)$ is independent of b given (a, v). Hence,

conditioned on the observed (a, v), the best possible data-only predictor for b has advantage

$$\left| \Pr[y = b \mid a, v] - \tfrac{1}{2} \right| \;=\; \tfrac{1}{2} \, |1 - 2\rho|^{m(a)}, \quad m(a) := \sum_{j \le g} a_j,$$

so the leakage decreases exponentially in the number of active prefix bits.

Under the oracle, $a_1, \ldots, a_g \overset{i.i.d.}{\sim} \text{Bernoulli}(1/2)$, so $m(a) \sim \text{Bin}(g, 1/2)$ and a typical example has $m(a) \approx g/2$. Therefore, when ρ is chosen close to $1/2$, a typical sample provides essentially no information about b without using the revealed prefix. In the ideal balanced case $\rho(w^\star) = 1/2$, we have $|1 - 2\rho| = 0$, so the bias is exactly 0 whenever $m(a) \geq 1$: the label is then a perfect one-time pad for b. The only trivial leakage is the rare event $m(a) = 0$ (no active prefix bit), in which case $B(a, v) = 0$ and $y = b$. Since $\Pr[m(a) = 0] = 2^{-g}$, the probability of seeing at least one such leakage example in K samples is $1 - (1 - 2^{-g})^K \approx K 2^{-g}$ when 2^{-g} is small. In our benchmark we simply reject and resample until $m(a) \geq 1$ to remove this degenerate case.

History-only baseline. A second shortcut is to ignore the step-specific evidence S_g and predict the next term using only the revealed prefix $P_g = (t_1, \ldots, t_g)$. In our construction, the prefix reveals exactly the previously sampled payload supports $(S_1, \ldots, S_g)$ (and their order), but under the instance distribution it carries no information about the next support S_{g+1}. Concretely, when supports are sampled i.i.d. uniformly *with replacement* from $\{S \subseteq [p] : |S| = d - 1\}$, we have $S_{g+1} \perp P_g$, so the conditional law of S_{g+1} given P_g remains uniform. Therefore any history-only strategy can do no better than prior guessing among the $\binom{p}{d-1}$ possible payload supports. The next lemma formalizes this and pins down the corresponding baseline success rate.

Lemma 4.3 (History-only is prior guessing)**.** *Assume the instance distribution samples supports $S_1, \ldots, S_n$ i.i.d. uniformly (*with replacement*) from $\{S \subseteq [p] : |S| = d - 1\}$. Then for any $g < n$, conditioned on the revealed prefix P_g, the next support S_{g+1} is uniform over $\{S \subseteq [p] : |S| = d - 1\}$ and independent of P_g. Consequently, any history-only estimator satisfies $\Pr[\widehat{S} = S_{g+1}] \leq \frac{1}{\binom{p}{d-1}}$.*

Recoverability in polynomial time for diligent solvers. See Appendix C.

5 The Benchmark

We implement the theoretical construction in Sec. 4 as a procedurally generated dataset and evaluation pipeline that tests an LLM's *reasoning horizon*: how far it can reliably extend an evolving partial solution when the next step is hidden from either the data alone or the history alone.

5.1 Data generation: the adversarial oracle

Our generator acts as an **adversarial oracle**: it produces examples whose labels are information-theoretically uninformative unless the solver conditions on the revealed prefix, exactly (Sec. 4.2).

Each instance is parameterized by the total number of variables $N = n + p$, the number of steps (address bits) n, and the payload degree $d-1$. We use d for the *total* degree of each ANF term, so each term contains exactly one address variable and $d-1$ payload variables.

Instance synthesis. We sample an ordered sequence of payload supports $S_1, \ldots, S_n \subseteq [p]$ with $|S_j| = d-1$ and define the ANF terms $t_j(a, v) = a_j \prod_{i \in S_j} v_i$. This fixes the ground-truth target $f(a, v) = \bigoplus_{j=1}^{n} t_j(a, v)$ and the unique stepwise curriculum $\{t_1, \ldots, t_n\}$ for the instance.

Curriculum at step g. We evaluate the model over steps $g = 0, 1, \ldots, n-1$. At step g, the model is given the explicit algebraic prefix $P_g = (t_1, \ldots, t_g)$ and must predict the unique next term t_{g+1}.

Observation synthesis (fixed-prefix obfuscation sampling). For each step g, we generate a fresh labeled set $\mathsf{S}_g = \{(x^{(k)}, y^{(k)})\}_{k=1}^{K}$ using the oracle in Sec. 4.2: we set $a_{g+1} = 1$, set all future address bits $a_j = 0$ for $j > g+1$, sample prefix address bits $a_1, \ldots, a_g \overset{i.i.d.}{\sim} \text{Bernoulli}(1/2)$, and sample payloads v uniformly from the Hamming sphere $\{v : \|v\|_0 = w^\star\}$. We choose $w^\star$ so that the monomial firing probability $\rho(w) = \binom{w}{d-1} / \binom{p}{d-1}$ is close to $1/2$, making monomial evaluations nearly balanced.

Ensuring the step is decodable (no accidental degeneracy). To avoid trivial failures due to degenerate sampling (e.g., too few positive residuals or insufficiently informative payloads), we optionally reject and resample S_g until it satisfies simple decodability checks aligned with our decoder in Sec. 4.3 (e.g., $|K_+| \geq 1$ and the intersection-based recovery does not collapse to an empty/ambiguous set). This guarantees that a valid reasoning path exists for the step, while the labels remain obfuscated for solvers that ignore the prefix.

5.2 Interaction Protocol

We frame each instance as an iterative completion game. At step g, the model receives a prompt containing: (i) **Global metadata:** the total number of variables and the target degree (or degree bound); (ii) **Partial solution (P_g):** the current ANF prefix, written as an XOR of monomials (e.g., `x0*x5 + x1*x2`); (iii) **Observations (S_g):** a list of K=64 labeled examples, each formatted as a full assignment to $(x_0, \ldots, x_{N-1})$ together with the binary output.

The model must output *exactly one* new monomial (e.g., `x3*x7`). We parse and validate it as a single well-formed monomial over the available variables, and score it correct iff it matches the ground-truth next term t_{g+1}.

Internally, the generator distinguishes *address* and *payload* variables (Sec. 4.2). However, the prompt presents a flat list $\{x_0, \ldots, x_{N-1}\}$ to avoid hand-coded cues and to test whether the model can infer and exploit the latent structure from the prefix and examples alone.

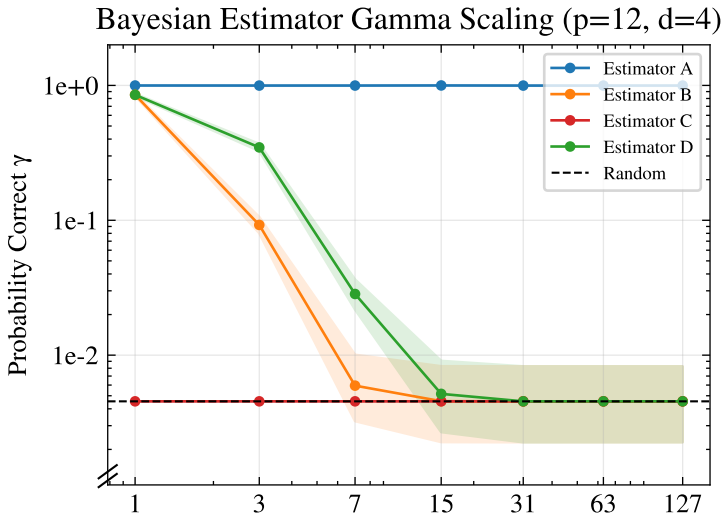

Figure 3: Only history+data sustains reliable next-step prediction. Step success γ_g (probability mass on the correct next monomial) versus depth g for each estimator class. Curves show the mean over 2000 generated circuits per depth, with shaded Jeffreys intervals. The diligent estimator $\mathcal{A}$ (history+data) maintains high γ_g across depths, whereas $\mathcal{B}$ (data-only) and $\mathcal{D}$ (partial) frequently collapse toward zero mass, and $\mathcal{C}$ (history-only) remains at chance.

5.3 Validator and Evaluation

Although the learner faces the hard problem of inferring the next monomial from data, validation is cheap. By construction, the monomial is barely identifiable and tail terms are disabled, so the only valid next term any algorithm could infer from the observed data is exactly the one we removed. Thus, validation is a constant-time check. This is a key advantage of our setup: in general circuit synthesis, verifying whether a proposed intermediate step can still lead to a correct final solution can require super-exponential time.

Within the Diligent Learner framework, γ lower-bounds the probability of choosing a correct next step, and task difficulty grows with depth. We estimate γ by computing the average step success γ_g at each depth (using depths that are powers of two) and then taking the minimum over depths. This quantifies a model's reasoning by identifying the depth where performance collapses and, by comparing to variants of estimator $\mathcal{D}$, infers the average fraction of the prefix the model can reliably reason about.

6 Results

We evaluated the four estimators described in Sec. 3.1, small LLMs, and frontier models to analyze empirically how gamma changes with depth.

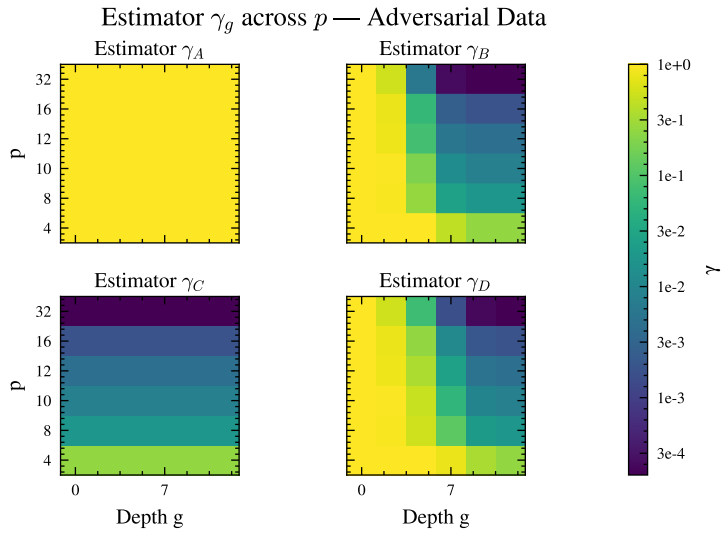

Figure 4: As both g and p increase, the probability of an estimator with imperfect information begins to collapse to zero. Only Estimator $\mathcal{A}$ is able to consistently produce the next monomial. The above heatmap was constructed through generating 200 circuits for each combination of hyperparameters and computing the corresponding γ_g for each p.

6.1 Estimator Simulations

We evaluated small LLMs to test whether they exhibit the same depth-induced degradation in next-step prediction as estimator $\mathcal{D}$. As shown in Figure 5, all models display a systematic decline in γ_g with depth, even though an explicit polynomial-time decoder exists at every step (Thm. C.1). Larger models and "thinking" variants perform better at shallow depths, but depth sensitivity persists.

We consider four models from the Qwen3-2507 family: `4B-Instruct`, `4B-Thinking`, `30B-A3B-Thinking`, and `30B-A3B-Instruct` [Team, 2025]. We run inference in vLLM on 3000 generated instances, evenly split across $g \in \{1, 3, 7, 15, 31\}$, using adversarial sampling with $p = 12$ and $d = 4$ [Kwon et al., 2023]. `4B-Instruct` does not achieve performance statistically distinguishable from random guessing even at the easiest setting, so we omit it for readability. `Qwen3-30B-A3B-Thinking` has a clear advantage at small depths over its instruct variant, but still drops sharply at intermediate depths (around g=15 here) and approaches the trivial baseline γ_{triv}.

6.2 Small LLMs

We evaluated small LLMs to test whether they exhibit the same depth-induced degradation in next-step prediction as estimator $\mathcal{D}$. As shown in Figure 5, all models display a systematic decline in γ_g with depth, qualitatively

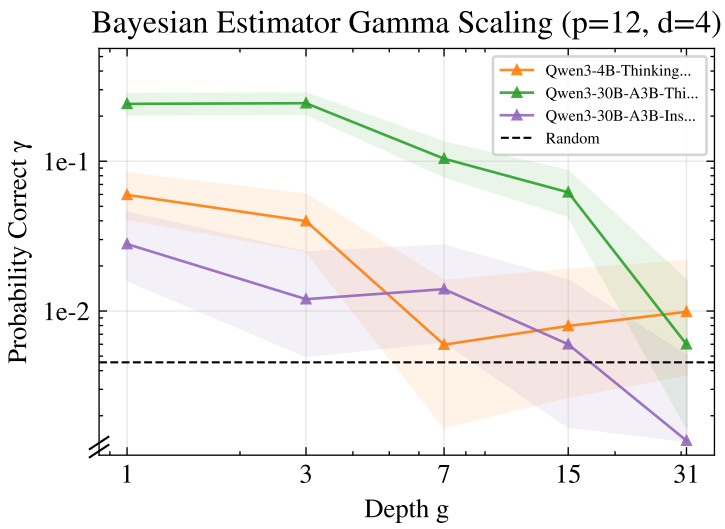

Figure 5: Small LLMs exhibit depth-induced collapse in next-step prediction. Step-success γ_g (probability mass on the correct next monomial) versus circuit depth g for Qwen3-2507 models under adversarial sampling ($p = 12$, $d = 4$). Despite the existence of a polynomial-time decoder at every step (Thm. C.1), all models degrade with depth: larger and "thinking" variants help at small g, but performance drops sharply at intermediate depths and approaches the trivial baseline γ_{triv}, indicating limited ability to maintain the prefix-conditioned cancellation required for continued progress.

mirroring the collapse seen for partial-information estimators, even though an explicit polynomial-time decoder exists at every step (Thm. C.1). Larger models and "thinking" variants perform better at shallow depths, but depth sensitivity persists.

We consider four models from the Qwen3-2507 family: `4B-Instruct`, `4B-Thinking`, `30B-A3B-Thinking`, and `30B-A3B-Instruct`. We run inference in vLLM on 3000 generated instances, evenly split across $g \in \{1, 3, 7, 15, 31\}$, using adversarial sampling with $p = 12$ and $d = 4$. This task requires long contexts for consistent decoding (max context length 32,768 for 4B and 81,920 for 30B). `4B-Instruct` does not achieve performance statistically distinguishable from random guessing even at the easiest setting, so we omit it for readability. `Qwen3-30B-A3B-Thinking` has a clear advantage at small depths over its instruct variant, but still drops sharply at intermediate depths (around g=15 here) and approaches the trivial baseline γ_{triv}.

Effective-prefix analysis (linking to partial-information estimators). To connect these empirical trends to the "partial access" thread (Sec. 4.1), we fit each model's accuracy curve to an effective-prefix abstraction: performance at depth g is modeled as if the solver only uses k of the g revealed terms, with either proportional scaling $k = ug$ or constant capacity $k = v$. Table 1 shows strong evidence that `Qwen3-30B-A3B-Thinking` behaves like it uses a constant fraction of the revealed prefix ($u \approx 0.77$), whereas `Qwen3-30B-A3B-Instruct` exhibits only weak scaling ($u \approx 0.15$). The 4B models do not meaningfully distinguish proportional and constant-capacity fits. This data suggests that as g grows, the oracle mask involves an expanding set of prefix monomials, and limited effective prefix utilization pushes models toward the partial-information regime.

6.3 Frontier LLMs

We now extend our analysis of depth-induced collapse in small models to larger systems: GPT 5.2 with extended Thinking, Claude Opus 4.5 with max Thinking, and Gemini 3 Pro (Jan 2026). Because frontier models typically generate extremely long reasoning traces on this task, it was financially prohibitive to replicate the full experimental protocol from Section 6.2. Nonetheless, the findings in the previous section provide a strong prior that guides our interpretation of the observed trends in γ_g for these larger models, even with fewer data points. In this section, we report results from 60 queries per model, spread across $g \in 31, 63, 127$ with $p = 12$ and $d = 4$. For half of the prompts, the model was explicitly instructed not to use tool calls; for the other half, it was free to choose any solution strategy, including tool use.

Frontier models are significantly better. Under the hardest conditions shown in 5, at which all small LLMs perform at random on the task, frontier models are still able to solve each step with a very high probability (Figure 6).

Tools stabilize γ_g over long horizons. Tool-enabled models maintain near-unity γ_g even at g=127, far exceeding the trivial baseline γ_{triv} (Figure 7).

These results offer a clear perspective on the Diligent Learner framework and on multi-step reasoning more broadly. Each reasoning step involves two distinct

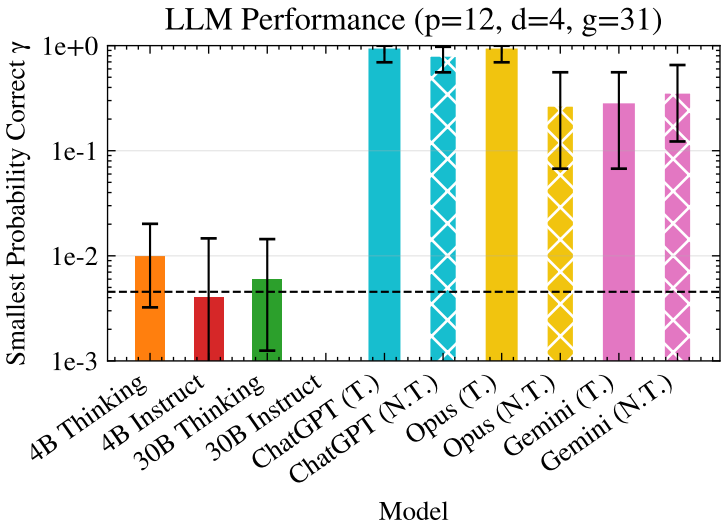

Figure 6: Frontier LLMs have a much higher γ_g than the smaller LLMs we tried.

requirements: inferring the correct constraints from the available inputs and data, and executing the computation implied by those constraints. Failure at either stage causes the reasoning chain to break.

Tool use fundamentally alters this dynamic. By externalizing execution, tools allow the model to focus primarily on specifying constraints rather than simultaneously discovering and implementing the full computation. This separation is critical for generalization. Prior work on LLM generalization suggests that, absent tools, success on out-of-distribution reasoning requires sparse compositional structure within the model's parameters, enabling both the representation of constraints and the implicit execution of the induced algorithm. This places a strong burden on the transformer's internal weights.

When tools are available, the model instead communicates constraints explicitly and delegates execution to an external program. This results in a much sparser effective algorithm at each step, which substantially improves generalization and stabilizes stepwise success probability. This mechanism explains why all tool-using models exhibit dramatically higher and more stable γ_g than their no-tool counterparts.

Tool-based reasoning remains imperfect, however, as it still relies on copying intermediate data through context. A natural extension would allow models to apply learned programs directly to their inputs, avoiding this degradation. Notably, the only model to perform well without explicit tool use was Opus, but closer inspection suggests that it frequently invoked tool-like behavior despite instructions to the contrary, though such usage was not always transparent.

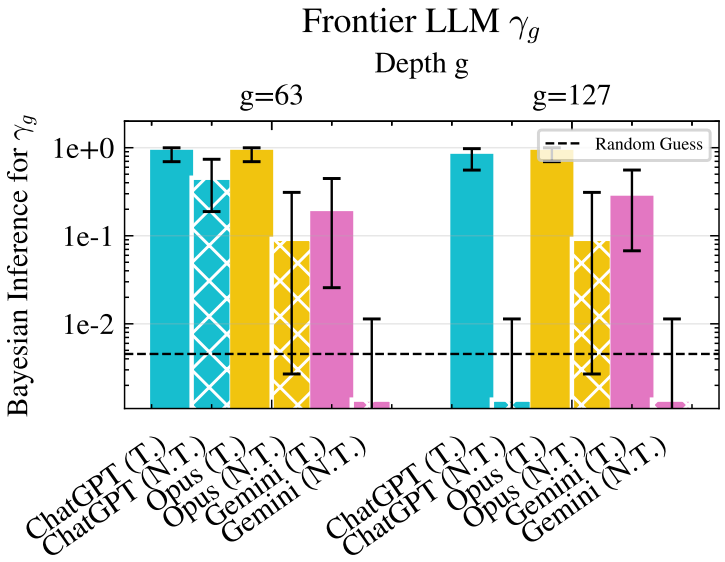

Figure 7: Frontier models that use tool calls (denoted with T.) have a much larger γ_g for this task and see minimal degradation, even at very significant model depths. When models are instructed not to use tools (denoted by N.T.), performance drops substantially as problem size increases. Opus often still used tool calls even when instructed not to, leading to inflation of its no-tools score, though it was challenging to determine exactly which instances used tools. The bars show the Bayesian confidence interval with a random prior as supported by the results in Section 6.2.

7 Discussion

In this work, we offer a rigorous empirical test of the *Diligent Learner* hypothesis by introducing a GF(2) circuit reconstruction benchmark that is explicitly adversarial to common shortcut strategies. The task forces a model to maintain state and repeatedly fuse accumulated historical context with newly observed evidence at every step, rather than relying on shallow pattern matching. Across this benchmark, we observe a clear divide in behavior: smaller language models exhibit a superlinear decline in γ as problem depth increases, effectively acting as partial-information estimators that cannot preserve the needed state. In contrast, frontier models that leverage tools maintain high γ over long sequences by delegating state tracking and verification to external mechanisms. Within the Diligent Learner framework, this suggests that progress toward "superintelligence" depends less on scaling test-time compute or deepening search, and more on architectures that can build and use tools.

Impact Statement

This work is theoretical. It raises no direct ethical, safety, or environmental concerns. We study conditions under which the Diligent Learner framework fails, introduce a benchmark that exposes this failure mode, and provide empirical evidence that stepwise reasoning degrades with depth in current models. We also show that explicit tool construction can mitigate this degradation by stabilizing long-horizon reasoning. These results inform the design and evaluation of future reasoning systems but do not introduce immediate real-world risks.

References

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL `https://arxiv.org/abs/2110.14168`.

Simon Frieder, Jonas Bayer, Sam Looi, Jacob Loader, Julius Berner, Katherine M. Collins, András Juhász, Fabian Ruehle, Sean Welleck, Gabriel Poesia, Ryan-Rhys Griffiths, Adrian Weller, Anirudh Goyal, Cameron Freer, Thomas Lukasiewicz, and Timothy Gowers. Data for mathematical copilots: Better ways of presenting proofs for machine learning, 2025. URL `https://arxiv.org/abs/2412.15184`.

Yao Fu, Hao Peng, Litu Ou, Ashish Sabharwal, and Tushar Khot. Specializing smaller language models towards multi-step reasoning. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*,

pages 10421–10430. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/fu23d.html.

John Garrett, Echedey Luis, H.-H. Peng, Tim Cera, gobinathj, Josh Borrow, Mehmet Keçeci, Splines, Suraj Iyer, Yuming Liu, cjw, and Mikhail Gasanov. garrettj403/scienceplots: 2.1.1, November 2023. URL https://zenodo.org/records/10206719.

Shibo Hao, Tianyang Liu, Zhen Wang, and Zhiting Hu. ToolkenGPT: Augmenting frozen language models with massive tools via tool embeddings. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=BHXsb69bSx.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R Narasimhan. SWE-bench: Can language models resolve real-world github issues? In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=VTF8yNQM66.

Nirmit Joshi, Gal Vardi, Adam Block, Surbhi Goel, Zhiyuan Li, Theodor Misiakiewicz, and Nathan Srebro. A theory of learning with autoregressive chain of thought, 2025. URL https://arxiv.org/abs/2503.07932.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles (SOSP '23)*, 2023. doi: 10.1145/3600006.3613165.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. Agentbench: Evaluating llms as agents, 2025. URL https://arxiv.org/abs/2308.03688.

Eran Malach. Auto-regressive next-token predictors are universal learners. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. Show your work: Scratchpads for intermediate computation with language models, 2021. URL https://arxiv.org/abs/2112.00114.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Aaron Parisi, Yao Zhao, and Noah Fiedel. Talm: Tool augmented language models, 2022. URL https://arxiv.org/abs/2205.12255.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, dahai li, Zhiyuan Liu, and Maosong Sun. ToolLLM: Facilitating large language models to master 16000+ real-world APIs. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=dHng2O0Jjr.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=Yacmpz84TH.

Shai Shalev-Shwartz and Amnon Shashua. From reasoning to super-intelligence: A search-theoretic perspective. *arXiv preprint arXiv:2507.15865*, 2025a. URL https://arxiv.org/abs/2507.15865.

Shai Shalev-Shwartz and Amnon Shashua. From reasoning to super-intelligence: A search-theoretic perspective. *arXiv preprint arXiv:2507.15865*, 2025b.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Zhengliang Shi, Shen Gao, Lingyong Yan, Yue Feng, Xiuyi Chen, Zhumin Chen, Dawei Yin, Suzan Verberne, and Zhaochun Ren. Tool learning in the wild: Empowering language models as automatic tool agents. In *Proceedings of the ACM on Web Conference 2025*, WWW '25, page 2222–2237, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400712746.

doi: 10.1145/3696410.3714825. URL https://doi.org/10.1145/3696410.3714825.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *arXiv preprint arXiv:2303.11366*, 2023. doi: 10.48550/arXiv.2303.11366. URL https://arxiv.org/abs/2303.11366.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning Text and Embodied Environments for Interactive Learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. URL https://arxiv.org/abs/2010.03768.

Shivam Singhal, Eran Malach, Tomaso Poggio, and Tomer Galanti. LLM-ERM: A probabilistic framework for in-context learning and text generation. *arXiv preprint arXiv:2510.14331*, 2025.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. Challenging big-bench tasks and whether chain-of-thought can solve them, 2022. URL https://arxiv.org/abs/2210.09261.

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022. URL https://arxiv.org/abs/2203.11171.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022. doi: 10.48550/arXiv.2201.11903. URL https://arxiv.org/abs/2201.11903.

Chenxiao Yang, Zhiyuan Li, and David Wipf. Chain-of-thought provably enables learning the (otherwise) unlearnable. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=N6pbLYLeej.

John Yang, Carlos E Jimenez, Alex L Zhang, Kilian Lieret, Joyce Yang, Xindi Wu, Ori Press, Niklas Muennighoff, Gabriel Synnaeve, Karthik R Narasimhan, Diyi Yang, Sida Wang, and Ofir Press. SWE-bench multimodal: Do AI systems generalize to visual software domains? In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=riTiq3i21b.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*, 2023. URL https://arxiv.org/abs/2305.10601.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. Webarena: A realistic web environment for building autonomous agents, 2024. URL https://arxiv.org/abs/2307.13854.

A Proofs

Lemma 4.1 (Monomial firing probability at fixed Hamming weight)**.** *Fix integers $p \geq d-1 \geq 1$ and $w \in \{0,\dots,p\}$. Let v be uniform over the Hamming sphere $\{v \in \{0,1\}^p : \|v\|_0 = w\}$, and fix any $S \subseteq [p]$ with $|S| = d-1$. Define $M_S(v) := \prod_{i\in S} v_i$. Then*

$$\Pr\big[M_S(v)=1\big] = \begin{cases} \frac{\binom{w}{d-1}}{\binom{p}{d-1}} = \frac{\binom{p-(d-1)}{w-(d-1)}}{\binom{p}{w}} & \text{if } w \geq d-1, \\ 0 & \text{if } w < d-1. \end{cases}$$

Proof. Write $\mathrm{supp}(v) := \{i \in [p] : v_i = 1\}$, so $|\mathrm{supp}(v)| = w$ and $\mathrm{supp}(v)$ is uniform over all $\binom{p}{w}$ subsets of $[p]$ of size w. Note that $M_S(v) = 1$ if and only if $v_i = 1$ for every $i \in S$, which is equivalent to $S \subseteq \mathrm{supp}(v)$. If $w < d-1$, no set of size w can contain S, hence $\Pr[M_S(v)=1] = 0$.

Assume now that $w \geq d-1$. The number of supports $T \subseteq [p]$ with $|T| = w$ that contain S is the number of ways to choose the remaining $w-(d-1)$ elements of T from the $p-(d-1)$ coordinates in $[p] \setminus S$, namely $\binom{p-(d-1)}{w-(d-1)}$. Since all $\binom{p}{w}$ supports are equally likely,

$$\Pr[M_S(v)=1] \;=\; \frac{\binom{p-(d-1)}{w-(d-1)}}{\binom{p}{w}} \;=\; \frac{\binom{w}{d-1}}{\binom{p}{d-1}}.$$

□

Lemma 4.2 (Bayes masking given observed (a,v))**.** *Assume the instance distribution samples $S_1,\dots,S_g,S_{g+1}$ i.i.d. uniformly from $\{S \subseteq [p] : |S| = d-1\}$, independently of the oracle samples. Fix a step g and condition on a realized example (a,v) with $\|v\|_0 = w^\star$. Let*

$$\rho := \rho(w^\star) = \Pr_S\big[M_S(v)=1\big] = \frac{\binom{w^\star}{d-1}}{\binom{p}{d-1}}, \quad m := m(a).$$

Then, marginalizing over the unknown prefix supports $(S_1,\dots,S_g)$, for each $r \in \{0,1\}$,

$$\Pr\big[B(a,v) = r \mid a, v\big] \;=\; \tfrac{1}{2}\big[1 + (-1)^r(1-2\rho)^m\big].$$

Moreover, $B(a,v)$ is independent of $b = M_{g+1}(v)$ given (a,v), and since $y = B(a,v) \oplus b$ we have

$$\big|\Pr[y = b \mid a, v] - \tfrac{1}{2}\big| \;=\; \tfrac{1}{2}|1-2\rho|^m.$$

Proof. Condition on a realized (a,v) with $\|v\|_0 = w^\star$. For each $j \leq g$ with $a_j = 1$, the random support S_j is uniform over $\{S \subseteq [p] : |S| = d-1\}$. Define $X_j := \mathbf{1}\{S_j \subseteq \mathrm{supp}(v)\}$. Then $X_j \sim \mathrm{Bernoulli}(\rho)$ with

$$\Pr[X_j = 1 \mid a, v] = \frac{\binom{w^\star}{d-1}}{\binom{p}{d-1}} =: \rho,$$

since exactly $\binom{w^\star}{d-1}$ of the $(d-1)$-subsets of $[p]$ lie inside the fixed set $\mathrm{supp}(v)$ of size $w^\star$. Because $S_1, \ldots, S_g$ are i.i.d., the collection $\{X_j : a_j = 1\}$ consists of $m := \sum_{j \le g} a_j$ independent Bernoulli(ρ) bits. Therefore $B(a, v) = \bigoplus_{j:a_j=1} X_j$ is the parity of m i.i.d. Bernoulli(ρ) bits, and the standard parity identity yields

$$\Pr[B(a,v) = 1 \mid a, v] = \frac{1 - (1-2\rho)^m}{2}, \qquad \Pr[B(a,v) = 0 \mid a, v] = \frac{1 + (1-2\rho)^m}{2}.$$

Independence of $B(a, v)$ and $b = M_{g+1}(v)$ given (a, v) holds because $B(a, v)$ depends only on $(S_1, \ldots, S_g)$ while b depends only on S_{g+1}, and these supports are independent under the instance distribution. Finally, since $y = B \oplus b$, we have $y = b$ iff $B = 0$, yielding

$$\Pr[y = b \mid a, v] = \Pr[B(a,v) = 0 \mid a, v] = \frac{1 + (1-2\rho)^m}{2}.$$

□

Lemma 4.3 (History-only is prior guessing)**.** *Assume the instance distribution samples supports $S_1, \ldots, S_n$ i.i.d. uniformly (*with replacement*) from $\{S \subseteq [p] : |S| = d-1\}$. Then for any $g < n$, conditioned on the revealed prefix P_g, the next support S_{g+1} is uniform over $\{S \subseteq [p] : |S| = d-1\}$ and independent of P_g. Consequently, any history-only estimator satisfies $\Pr[\widehat{S} = S_{g+1}] \le \frac{1}{\binom{p}{d-1}}$.*

Proof. Let $\mathcal{U}$ be the uniform distribution over $\{S \subseteq [p] : |S| = d-1\}$. By assumption, $S_{g+1} \sim \mathcal{U}$ and $S_{g+1} \perp (S_1, \ldots, S_g)$. Since P_g is a deterministic function of $(S_1, \ldots, S_g)$, we also have $S_{g+1} \perp P_g$, hence the conditional law of S_{g+1} given P_g remains $\mathcal{U}$. Therefore any estimator based only on P_g succeeds with probability at most $\max_S \Pr[S_{g+1} = S] = 1/\binom{p}{d-1}$. □

B Other Simulations

C The Monomial can be Recovered in Polynomial Time

Given P_g, a solver can form residual labels

$$r^{(k)} \;:=\; y^{(k)} \oplus \bigoplus_{j=1}^{g} a_j^{(k)} M_j\big(v^{(k)}\big) \;=\; M_{g+1}\big(v^{(k)}\big), \tag{8}$$

so recovery reduces to identifying the unknown degree-$(d-1)$ support S_{g+1} from labeled payloads.

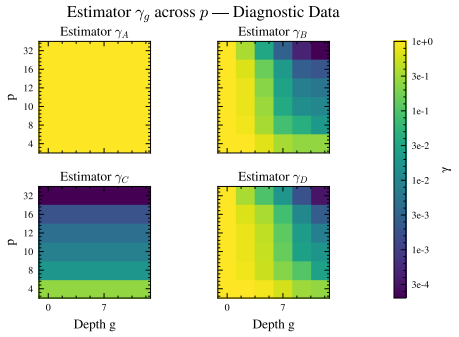

Figure 8: As both g and p increase, the probability of an estimator with imperfect information begins to collapse to zero. Only Estimator $\mathcal{A}$ is able to consistently produce the next monomial. The above heatmap was constructed through generating 200 circuits for each combination of hyperparameters and computing the corresponding γ_g for each p. This diagram shows the performance when not using the adversarial dataset construction.

Let $K_+ := \{k : r^{(k)} = 1\}$ and $T := |K_+|$. Consider the decoder

$$\widehat{S} \;:=\; \bigcap_{k \in K_+} \operatorname{supp}\big(v^{(k)}\big), \tag{9}$$

which outputs $\widehat{S}$ if $T \geq 1$ and $|\widehat{S}| = d - 1$, and otherwise declares failure.

Theorem C.1 (Poly-time recovery under fixed-weight payloads)**.** *Fix an instance and assume payloads are i.i.d. uniform on* $\{v \in \{0,1\}^p : \|v\|_0 = w^\star\}$ *with* $w^\star \geq d-1$. *Let* $\rho = \rho(w^\star) = \Pr[M_{g+1}(v) = 1]$ *and* $T = |K_+|$. *Then* $\Pr[T = 0] = (1-\rho)^K$. *Moreover, conditioned on* $T \geq 1$, *the decoder succeeds with probability at least*

$$1 - \min\Big\{1,\ (p-(d-1))\Big(\frac{w^\star - (d-1)}{p-(d-1)}\Big)^T\Big\}. \tag{10}$$

Proof. If $r^{(k)} = 1$ then $M_{g+1}\big(v^{(k)}\big) = 1$, which is equivalent to $S_{g+1} \subseteq \operatorname{supp}\big(v^{(k)}\big)$. Thus for every $k \in K_+$ we have $S_{g+1} \subseteq \operatorname{supp}\big(v^{(k)}\big)$, and hence

$$S_{g+1} \subseteq \bigcap_{k \in K_+} \operatorname{supp}\big(v^{(k)}\big) \;=\; \widehat{S}.$$

Therefore, conditioned on $T \geq 1$, the intersection decoder can fail only if $\widehat{S}$ contains at least one *extraneous* coordinate $i \notin S_{g+1}$, i.e., some $i \in [p] \setminus S_{g+1}$ appears in every positive support.

Fix any $i \notin S_{g+1}$ and consider a single draw v conditioned on $r = 1$ (equivalently, $S_{g+1} \subseteq \operatorname{supp}(v)$). Under the fixed-weight model, after forcing ones on S_{g+1}, the remaining $w^\star - (d-1)$ ones are chosen uniformly among the $p-(d-1)$

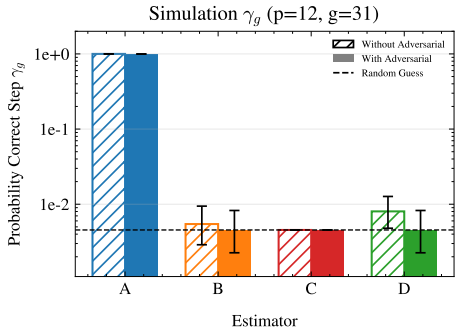

Figure 9: The figure above shows how γ_g changes for Estimators $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, and $\mathcal{D}$ when the data is biased adversarially (as in the results so far) preventing identification monomials from frequency statistics, and without de-biasing.

Table 1: **Likelihood-based fit of LLM accuracy to an effective-prefix model.** Accuracy as a function of prefix length g is fit using two one-parameter models for the effective number of known prefix terms: proportional $k = ug$ and constant-capacity $k = v$. Fits use the binomial log-likelihood aggregated across depths; models are compared via AIC. Reported $\Delta\text{AIC} = \text{AIC}_{\text{constant}} - \text{AIC}_{\text{proportional}}$, so positive values favor proportional scaling. Values $\Delta\text{AIC} < 2$ indicate no meaningful distinction.

Model	u	v	ΔAIC	Better
Qwen3-30B-A3B-Instruct-2507	0.15	0.00	2.21	u
Qwen3-30B-A3B-Thinking-2507	0.47	0.00	228.08	u
Qwen3-4B-Instruct-2507	0.08	0.00	2.32	u
Qwen3-4B-Thinking-2507	0.05	0.00	0.00	–

coordinates in $[p] \setminus S_{g+1}$. Hence

$$\Pr[i \in \text{supp}(v) \mid r = 1] \;=\; \frac{w^\star - (d-1)}{p - (d-1)}.$$

Now condition on the event $\{T = |K_+|\}$ and on the index set K_+ itself. Because the original examples are i.i.d., the payloads $\{v^{(k)}\}_{k \in K_+}$ are i.i.d. draws from the conditional distribution $(v \mid r = 1)$, so

$$\Pr\left[i \in \text{supp}(v^{(k)})\ \forall k \in K_+ \,\middle|\, T\right] \;=\; \left(\frac{w^\star - (d-1)}{p - (d-1)}\right)^T.$$

Taking a union bound over the $p - (d-1)$ possible extr coordinates gives

$$\Pr\left[\widehat{S} \neq S_{g+1} \mid T\right] \;\le\; \big(p - (d-1)\big)\left(\frac{w^\star - (d-1)}{p - (d-1)}\right)^T,$$

which implies (10). Finally, since $T = \sum_{k=1}^{K} \mathbf{1}\{r^{(k)} = 1\}$ with $\Pr[r^{(k)} = 1] = \rho = \rho(w^\star)$, we have $\Pr[T = 0] = (1-\rho)^K$. □

Corollary C.2 (High-probability recovery with K samples)**.** *In the setting of Thm. C.1, let* $\rho := \rho(w^\star) = \Pr[M_{g+1}(v) = 1]$ *and* $\alpha := \frac{w^\star-(d-1)}{p-(d-1)} \in [0,1)$*. Fix* $\delta \in (0,1)$*. For* $\alpha \in (0,1)$ *define*

$$T_0 := \left\lceil \frac{\log\big(2(p-(d-1))/\delta\big)}{\log(1/\alpha)} \right\rceil \quad (\textit{and if } \alpha = 0,\ \textit{take } T_0 := 1).$$

If $K \geq \frac{1}{\rho}\max\{2T_0,\ 8\log(2/\delta)\}$*, then the decoder in* (9) *outputs* $\widehat{S} = S_{g+1}$ *with probability at least* $1-\delta$*.*

Proof. Let $T = |K_+| = \sum_{k=1}^{K} \mathbf{1}\{r^{(k)} = 1\}$. Since $r^{(k)} = M_{g+1}\big(v^{(k)}\big)$ and the payloads are i.i.d., we have $T \sim \mathrm{Bin}(K, \rho)$.

By Thm. C.1, for any $t \geq 1$,

$$\Pr[\widehat{S} \neq S_{g+1} \mid T = t] \ \leq\ \big(p-(d-1)\big)\alpha^t.$$

Hence

$$\Pr[\widehat{S} \neq S_{g+1}] \ \leq\ \Pr[T < T_0] \ +\ \Pr[\widehat{S} \neq S_{g+1} \mid T \geq T_0].$$

For the first term, our lower bound on K implies $K\rho/2 \geq T_0$, so by a multiplicative Chernoff bound,

$$\Pr[T < T_0] \ \leq\ \Pr\big[T \leq \tfrac{1}{2}K\rho\big] \ \leq\ \exp\big(-\tfrac{1}{8}K\rho\big) \ \leq\ \delta/2,$$

where the last inequality uses $K\rho \geq 8\log(2/\delta)$.

For the second term, on $T \geq T_0$ we have

$$\Pr[\widehat{S} \neq S_{g+1} \mid T \geq T_0] \ \leq\ \big(p-(d-1)\big)\alpha^{T_0} \ \leq\ \delta/2,$$

by the definition of T_0 (and trivially if $\alpha = 0$). Combining the two bounds yields $\Pr[\widehat{S} \neq S_{g+1}] \leq \delta$. □

Lem. 4.2 shows a *per-sample* obfuscation property: marginalizing over the unknown prefix supports, each labeled example provides only exponentially small Bayes advantage about the next-term signal unless one conditions on the revealed prefix. In the ideal balanced case $\rho(w^\star) = 1/2$, this advantage is 0 whenever $m(a) \geq 1$. In our implementation we resample $a_1, \ldots, a_g$ until $m(a) \geq 1$ to remove the trivial leakage case $m(a) = 0$. Lem. 4.3 rules out history-only shortcuts under the instance distribution, since S_{g+1} remains uniform given the revealed prefix.

Finally, Thm. C.1 (and Corollary C.2) show that a diligent solver can subtract the prefix mask to obtain residual labels $r^{(k)} = M_{g+1}(v^{(k)})$ and recover t_{g+1} in polynomial time from $K = O\big(\frac{1}{\rho(w^\star)}\log(p/\delta)\big)$ samples with failure probability at most δ (for fixed d and constant $\rho(w^\star)$). Together, these properties justify using γ_g in (4) as an operational measure of step success in our benchmark.

Efficient validation. Given the instance specification (in particular S_{g+1}) and a candidate monomial $\tilde{t}$, the validator parses $\tilde{t}$, rejects unless it contains exactly one address variable (which must be a_{g+1}) and exactly $d-1$ distinct payload variables, then accepts iff the parsed payload index set $\widetilde{S}$ equals S_{g+1}.This runs in time $O(|\tilde{t}| + d \log d)$ worst-case (or $O(|\tilde{t}| + d)$ expected with hashing).